

2 VARIABLE



2.1 Variable Declaration + Use

2.2 Types

2.3 Scope

2.4 Interpolation



- Native CSS variable support is still in its infancy, but Sass affords us a way to set reusable values
- Variable names begin with \$, like \$base

2.1 Variable Declaration + Use



application.scss

```
$base: #777777;

.sidebar {
  border: 1px solid $base;
  p {
    color: $base;
  }
}
```

application.css

```
.sidebar {
  border: 1px solid #777777;
}
.sidebar p {
  color: #777777;
}
```

2.1 Variable Declaration + Use



The Default Flag

Variable definitions can optionally take the `!default` flag:

application.scss

```
$title: 'My Blog';  
$title: 'About Me';  
  
h2:before {  
  content: $title;  
}
```

overrides the
first value

application.css

```
h2:before {  
  content: 'About Me';  
}
```

2.1 Variable Declaration + Use



application.scss

```
$title: 'My Blog';  
$title: 'About Me' !default;
```

```
h2:before {  
  content: $title;  
}
```



Since a value exists,
it isn't overwritten



application.css

```
h2:before {  
  content: 'My Blog';  
}
```

2.1 Variable Declaration + Use



application.scss

```
$rounded: 5px;  
  
@import "buttons";
```

_buttons.scss

```
$rounded: 3px !default;  
  
.btn-a {  
  border-radius: $rounded;  
  color: #777;  
}  
  
.btn-b {  
  border-radius: $rounded;  
  color: #222;  
}
```

if a value isn't
defined elsewhere,
used by default

2.1 Variable Declaration + Use



application.scss

```
$rounded: 5px;  
  
@import "buttons";
```

application.css

```
.btn-a {  
  border-radius: 5px;  
  color: #777;  
}  
.btn-b {  
  border-radius: 5px;  
  color: #222;  
}
```

2.1 Variable Declaration + Use



Booleans

```
$rounded: false;  
$shadow: true;
```

Numbers - can be set with or without units:

```
$rounded: 4px;  
$line-height: 1.5;  
$font-size: 3rem;
```

2.2 Types



Colors

```
$base: purple;  
$border: rgba(0, 255, 0, 0.5);  
$shadow: #333;
```

Strings - can be set with or without quotes:

```
$header: 'Helvetica Neue';  
$callout: Arial;  
$message: "Loading...";
```

2.2 Types



Lists

```
$authors: nick, dan, aimee, drew;  
$margin: 40px 0 20px 100px;
```

Null

```
$shadow: null;
```



2.1 Variable Declaration + Use



2.2 Types



2.3 Scope



2.4 Interpolation



application.scss

```
p {  
  $border: #ccc;  
  border-top: 1px solid $border;  
}  
h1 {  
  border-top: 1px solid $border;  
}
```


\$border isn't available
outside of p

application.css

```
Syntax error: Undefined  
variable: "$border"
```

2.3 Scope



Reassignment in a Declaration

- Variables **set** inside a declaration (within { }) aren't usable outside that block
- Setting **new** values to variables set outside a declaration changes future instances



application.scss

```
$color-base: #777777;

.sidebar {
  $color-base: #222222;
  background: $color-base;
}

p {
  color: $color-base;
}
```

application.css

```
.sidebar {
  background: #222222;
}

p {
  color: #222222;
}
```

overwriting a variable in a
declaration is global

2.3 Scope



Use the Ruby-esque `#{$variable}` to shim variables into selectors, property names, and strings:

application.scss

```
$side: top;

sup {
  position: relative;
  #{$side}: -0.5em;
}

.callout-#{$side} {
  background: #777;
}
```

application.css

```
sup {
  position: relative;
  top: -0.5em;
}

.callout-top {
  background: #777;
}
```

2.4 Interpolation



Assembly Tip

Be considerate of variable naming. `$color-base` gets a lot more mileage than `$color-blue`.



ASSEMBLING SASS



PROPERTY OF:
CODE SCHOOL

PROJECT DATE:
SEP, 2012



This project was created and developed by
David Espinoza, Matthew S. Walsh,
and is used by permission of the
Assembly School Project. It is not to be
reproduced or distributed in any form
without the written consent of the
Assembly School Project.

S

STANDARD
INSPECTION : NO 18